



# UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE  
United States Patent and Trademark Office  
Address: COMMISSIONER FOR PATENTS  
P.O. Box 1450  
Alexandria, Virginia 22313-1450  
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/663,665	09/15/2000	Gregory L. Slaughter	5181-47300	2188

7590 10/18/2006

Robert C Kowert  
Conley Rose & Tayon P C  
P O Box 398  
Austin, TX 78767-0398

EXAMINER
----------

ZHEN, LI B

ART UNIT	PAPER NUMBER
----------	--------------

2194

DATE MAILED: 10/18/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

<b>Office Action Summary</b>	<b>Application No.</b> 09/663,665	<b>Applicant(s)</b> SLAUGHTER ET AL.	
	<b>Examiner</b> Li B. Zhen	<b>Art Unit</b> 2194	

**-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --**

**Period for Reply**

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

**Status**

- 1) ☒ Responsive to communication(s) filed on 04 August 2006.
- 2a) ☒ This action is **FINAL**.                      2b) ☐ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

**Disposition of Claims**

- 4) ☒ Claim(s) 1,3-22,24-43 and 45-53 is/are pending in the application.
- 4a) Of the above claim(s) \_\_\_\_\_ is/are withdrawn from consideration.
- 5) ☐ Claim(s) \_\_\_\_\_ is/are allowed.
- 6) ☒ Claim(s) 1,3-22,24-43 and 45-53 is/are rejected.
- 7) ☐ Claim(s) \_\_\_\_\_ is/are objected to.
- 8) ☐ Claim(s) \_\_\_\_\_ are subject to restriction and/or election requirement.

**Application Papers**

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on \_\_\_\_\_ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.  
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).  
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

**Priority under 35 U.S.C. § 119**

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All    b) ☐ Some    \* c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
2. ☐ Certified copies of the priority documents have been received in Application No. \_\_\_\_\_.
3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

\* See the attached detailed Office action for a list of the certified copies not received.

**Attachment(s)**

- 1) ☐ Notice of References Cited (PTO-892)
- 2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
- 3) ☐ Information Disclosure Statement(s) (PTO/SB/08)  
Paper No(s)/Mail Date \_\_\_\_\_.

- 4) ☐ Interview Summary (PTO-413)  
Paper No(s)/Mail Date. \_\_\_\_\_.
- 5) ☐ Notice of Informal Patent Application
- 6) ☐ Other: \_\_\_\_\_.

  
**WILLIAM THOMSON**  
**SUPERVISORY PATENT EXAMINER**

### **DETAILED ACTION**

1. Claims 1, 3-22, 24-43 and 45-53 are pending in the current application.

#### ***Response to Arguments***

2. Applicant's arguments filed 08/04/2006 have been fully considered but they are not persuasive. In response to the Non-Final Office Action dated 05/08/2006, applicant argues:

(1) Erickson does not mention anything about a client device receiving a message that includes a data representation language of a computer programming language object from a service device in a distributed computing environment. Reading a serialized wrapper file locally is quite different from receiving a message from another device that includes a data representation language representation of a computer programming language object. [p. 2]

(2) Erickson does not teach that serialization is used to send the wrapper file as a message (via RMI or any other technique) from a service device to a client device. [pp. 3 – 4]

(3) Erickson is not concerned with, nor does Erickson mention anything about, security or preventing unauthorized access to objects. The fact that Germscheid teaches deleting an object to prevent unauthorized access does not have any bearing on the teachings of Erickson or provide any motivation to modify Erickson's teachings. [pp. 4 – 5]

(4) Examiner also states that it would be obvious to modify Erickson according to the teachings of Germscheid because deleting the object "deallocates the storage for the object after the user has finished accessing the object." This motivation is not supported by any evidence of record and seems to just be the Examiner's own opinion formed in hindsight. Neither Germscheid nor Erickson mentions anything about deallocating storage. Moreover, as noted above, Erickson teaches that his wrapper should persist for subsequent use (see, e.g., Abstract, last sentence). Thus, Erickson actually teaches away from the Examiner's proposed combination. [pp. 5 – 6]

(5) The cited passage does not describe making a temporary copy of a wrapper file, as the Examiner suggests. Instead, Erickson states, "a wrapper execution engine ... can read the serialized wrapper (e.g. from the wrapper file), reproduce the wrapper within its execution environment, and run it (the wrapper)" (parenthesis added). [p. 6]

(6) Examiner's combination of Erickson and Germscheid fails to teach deleting, in response to a user terminating accessing a client device, a computer programming language object that was generated from a data representation language representation of the object. [p. 6-8]

(7) Erickson teaches that a wrapper file is read, such as from a local disk file as illustrated in figures 14 and 15. Erickson does not mention anything about a client device receiving a message that includes a data representation language of a computer programming language object from a service device in a distributed computing environment [p. 9]

(8) Examiner's combination of Erickson and Germscheid does not result in a system that teaches or suggests generating a computer programming language object from a data representation language representation of the object if it is determined that the user has access rights to the computer programming language object. [p.9]

(9) Erickson in view of Germscheid fails to teach or suggest wherein the message further includes access information for the computer programming language object, wherein said determining if the user has access rights to the computer programming object uses the access information.

(10) Germscheid, whether considered singly or in combination with Erickson, fails to teach or suggest anything regarding including accessing information for a computer programming language object in a message that also include a data representation language representation of the computer programming language object [pp. 11-12]

(11) Examiner's combination of Erickson, Germscheid and Wu fails to teach or suggest storing the computer programming language object in response to the user terminating accessing the client device. [p. 12]

(12) Germscheid teaches the deletion of objects in response to a user terminating access. Thus, Germscheid teaches away from storing a computer

Art Unit: 2194

programming language object in response to a user terminating accessing a client device [p. 13].

(13) Regarding claim 18, the Examiner's combination of Erickson, Germerscheid and Wu fails to teach or suggest storing access rights information of the user with the object, wherein accessing the stored object comprises verifying the access rights of a user with the stored access rights information. [p. 18]

In response to argument (1), examiner respectfully disagrees and submits that Erickson does not specify reading the wrapper file locally. Erickson clearly discloses a wrapper serialization component that provides for the storage and retrieval of wrappers in XML through the process of Object Serialization and serialization is used for communications via sockets or remote method invocation [col. 25, line 56 – col. 26, line 15]. The wrapper files [message] stored as an XML is retrieved from wrapper serialization component [service device] via sockets or remote method invocation [distributed computing environment]. Therefore, Erickson teaches receiving a message that includes a data representation language of a programming language object from a service device in a distributed computing environment.

In response to argument (2), examiner respectfully disagrees and submits that Erickson teaches sending the wrapper file as a message from a service device to a client. Erickson clearly discloses a wrapper serialization component that provides for the storage and retrieval of wrappers in XML through the process of Object Serialization and serialization is used for communications via sockets or remote method invocation [col. 25, line 56 – col. 26, line 15]. The wrapper files [message] stored as an XML is retrieved from wrapper serialization component [service device] via sockets or remote method invocation [distributed computing environment]. Therefore, Erickson teaches receiving a message that includes a data representation language of a programming language object from a service device in a distributed computing environment.

In response to argument (3), examiner respectfully disagrees and submits that a person of ordinary skill in the art would know that deleting an object will remove the object from the shared memory of the computer and prevent unauthorized access to the

Art Unit: 2194

objects. Preventing unauthorized access to objects would prevent users of Erickson's system from accessing data that they are not authorized.

As to argument (4), examiner respectfully disagrees and submits one of ordinary skilled in the art would know to delete an object and reclaim the memory for the object. In addition, Erickson discloses the Java environment [col. 3, lines 25 – 36] and garbage collection is a feature of Java. Garbage collection is a form of automatic memory management that reclaims memory used by objects that will never again be accessed. Therefore, the combination of Erickson and Germscheid would provide support for object persistence for objects that are needed in the future and reclaiming memory for objects that are no longer needed.

In response to arguments (5), examiner respectfully disagrees because Erickson discloses storing and retrieving a wrapper file [col. 20, lines 22-48]. When the wrapper file is read from storage a copy of the wrapper file would be created [read the serialized wrapper, reproduce the wrapper; col. 26, lines 20-30, emphasis added].

As to argument (6), examiner respectfully disagrees and notes that the security session objects of Germscheid corresponds to the wrapper objects of Erickson because both objects provides access to a network resource [i.e. col. 27, lines 52-60 of Erickson and col. 9, lines 22-30 of Germscheid].

In response to argument (7), examiner respectfully disagrees and submits that Erickson does not specify reading the wrapper file locally. Erickson clearly discloses a wrapper serialization component that provides for the storage and retrieval of wrappers in XML through the process of Object Serialization and serialization is used for communications via sockets or remote method invocation [col. 25, line 56 – col. 26, line 15]. The wrapper files [message] stored as an XML is retrieved from wrapper serialization component [service device] via sockets or remote method invocation [distributed computing environment]. Therefore, Erickson teaches receiving a message that includes a data representation language of a programming language object from a service device in a distributed computing environment.

As to argument (8), examiner respectfully disagrees and notes that the security session objects of Germscheid corresponds to the wrapper objects of Erickson because

Art Unit: 2194

both objects provides access to a network resource [i.e. col. 27, lines 52-60 of Erickson and col. 9, lines 22-30 of Germscheid]. Therefore, the combination of Germscheid and Erickson generates a computer programming language object if it is determined that the user has access rights to the computer programming language object.

In response to argument (9), examiner respectfully disagrees and submits that Germscheid teaches the service levels as defined by security profiles regulate access to a resource [i.e. col. 7, line 60-col. 8, line 20 of Germscheid]. Therefore, the service levels correspond to the recited access information because the service levels regulate access to resources.

In response to argument (10), examiner respectfully disagrees and submits that Erickson teaches sending the wrapper file as a message from a service device to a client. Erickson clearly discloses a wrapper serialization component that provides for the storage and retrieval of wrappers in XML through the process of Object Serialization and serialization is used for communications via sockets or remote method invocation [col. 25, line 56 – col. 26, line 15]. The wrapper files [message] stored as an XML is retrieved from wrapper serialization component [service device] via sockets or remote method invocation [distributed computing environment]. Therefore, Erickson teaches receiving a message that includes a data representation language of a programming language object from a service device in a distributed computing environment.

As to argument (11), examiner respectfully disagrees and notes that Erickson teaches the user terminating the accessing the client device and storing the computer programming language object in response to the terminating access [wrapper file has been created and stored; col. 26, lines 21 – 30, see also rejection to claim 16 below].

In response to argument (12), examiner respectfully disagrees and submits that Erickson teaches storing a computer programming language object in response to a user terminating accessing a client device [col. 26, lines 21 – 30]. Examiner notes that the claims recites both storing and deleting a computer programming language object in response to a user terminating accessing to a client device in separate dependent claims. Applicant is claiming both features but are arguing them separately on the basis that one feature would teach away from the other. Examiner submits that the

Art Unit: 2194

combination of Erickson and Germscheid is proper because this creates a flexible garbage collection policy such that objects that are needed in the future are not deleted and objects that are not needed in the future are deleted and the memory of the deleted object is reclaimed.

In response to argument (13), examiner respectfully disagrees and submits that Germscheid teaches the service levels as defined by security profiles regulate access to a resource [i.e. col. 7, line 60-col. 8, line 20 of Germscheid]. Therefore, the service levels correspond to the recited access information because the service levels regulate access to resources and accessing the stored object comprises verifying the access rights of the user with the stored access rights information [col. 7, line 60 – col. 8, line 3 of Germscheid].

### ***Claim Rejections - 35 USC § 103***

3. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

4. **Claims 1, 6-12, 16-22, 27-34, 40-43, 45-49 and 51-53 are rejected under 35 U.S.C. 103(a) as being unpatentable over U.S. Patent No. 6,851,089 to Erickson et al. [hereinafter Erickson] in view of U.S. Patent No. 6,782,425 to Germscheid et al. [hereinafter Germscheid, both references cited in the previous office action].**

5. As to claim 1, Erickson teaches the invention substantially as claim including a method for the exchange of objects in a distributed computing environment [col. 25, line 57 – col. 26, line 14], comprising:

user accessing a client device [col. 7, lines 1 – 16]; and

generating a computer programming language object from a data representation language representation of the object [a wrapper file has been created and stored, the



Art Unit: 2194

wrapper file can be read by a wrapper builder application and deserialized, by known methods, to reproduce the objects that the wrapper comprises; col. 26, lines 20 – 30], wherein the object is an instance of a class in the computer programming language [wrapper builder application employs serialization to encode an internal object representation of a wrapper into XML format; col. 26, lines 16 - 20], and wherein the object is accessible for use during the accessing the client device [wrapper serialization component 1170 provides for the storage and retrieval of wrappers in XML (Extensible Markup Language) through the process of Object Serialization; col. 25, line 59 - col. 26, line 13], and the client device receiving a message [XML data can be saved as a wrapper file; col. 26, lines 15-20] in the data representation language [wrapper serialization component 1170 implements the functionality by which wrappers are stored and retrieved, called serialization; col. 20, lines 22-48] from a service device in the distributed computing environment prior to the generating a computer programming language object [Serialization is used for lightweight persistence and for communication via sockets or Remote Method Invocation (RMI); col. 26, lines 3-16], wherein the message includes the data representation language representation of the object [XML format; col. 26, lines 15 – 20].

6. Although Erickson teaches the invention substantially as claimed, Erickson does not teach deleting the computer programming language object in response to the terminating access.

However, Germscheid teaches secure access to sensitive data [col. 4, lines 42 – 53] and deleting a computer programming language object in response to the terminating access [termination of session; col. 16, lines 1 – 6] so that the deleted object is not accessible by subsequent users of the client device [deletes the CCISession object which provides the secure access; col. 16, lines 1 – 6].

7. It would have been obvious to a person of ordinarily skilled in the art at the time of the invention to apply the teaching of deleting the computer programming language object in response to the terminating access as taught by Germscheid to the invention of Erickson because this would prevent unauthorized access to the object [col. 16, lines

Art Unit: 2194

4 – 5 of Germscheid] and deallocates the storage for the object after the user has finished accessing the object.

8. As to claim 6, Erickson as modified teaches generating a computer programming language object from a data representation language representation of the object is performed by a virtual machine executing within the client device [col. 25, lines 58 – 67 of Erickson; examiner notes that a virtual machine is inherent to the Java environment].

9. As to claim 7, Erickson as modified teaches generating a plurality of computer programming language objects from data representation language representations of the objects [col. 25, line 59 - col. 26, line 13 of Erickson], and deleting the plurality of computer programming language objects in response to the terminating access [col. 16, lines 1 – 6 of Germscheid].

10. As to claim 8, Erickson as modified teaches the data representation language is extensible Markup Language (XML) [XML; col. 25, line 59 – col. 26, line 13 of Erickson].

11. As to claim 9, Erickson as modified teaches the computer programming language is the Java programming language [col. 25, line 59 – col. 26, line 13 of Erickson].

12. As to claim 10, Erickson as modified teaches a method for the secure exchange of objects in a distributed computing environment [col. 25, line 57 – col. 26, line 14 of Erickson], comprising:

a user accessing a client device [col. 7, lines 1 – 16 of Erickson];

the client device receiving a message in a data representation language from a service device in the distributed computing environment [col. 16, lines 1 – 30 of Erickson], wherein the message includes a data representation language representation of an object [XML format; col. 26, lines 15 – 20 of Erickson];

if the determining, determines the user has access rights to the computer programming language object [col. 15, lines 38 – 48 of Germscheid], generating the

object from the data representation language representation of the object [a wrapper file has been created and stored, the wrapper file can be read by a wrapper builder application and deserialized, by known methods, to reproduce the objects that the wrapper comprises; col. 26, lines 20 – 30 of Erickson], wherein the object is an instance of a class in the computer programming language [wrapper builder application employs serialization to encode an internal object representation of a wrapper into XML format; col. 26, lines 16 – 20 of Erickson], and wherein the object is accessible for use during the accessing the client device [col. 25, line 59 - col. 26, line 13 of Erickson]; and

if the determining determines the user does not have access rights to the computer programming language object, not generating the object [col. 8, lines 4 – 11 of Germscheid].

13. As to claim 11, Erickson as modified teaches the message further includes access information for the computer programming language object, wherein the determining if the user has access rights to the computer programming language object uses the access information [col. 7, lines 38 – 50 and col. 7, line 60 – col. 8, line 21 of Germscheid].

14. As to claim 12, Erickson as modified teaches deleting the computer programming language object in response to the user terminating access to the client device, wherein the deleted object is not accessible for use by subsequent users of the client device [col. 16, lines 1 – 6 of Germscheid].

15. As to claim 16, Erickson as modified teaches the user terminating the accessing the client device and storing the computer programming language object in response to the terminating access [wrapper file has been created and stored; col. 26, lines 21 – 30 of Erickson].

16. As to claim 17, Erickson as modified teaches the user accessing the client device subsequent to the storing the object and accessing the stored object during the accessing the client device [col. 26, lines 21 – 30 of Erickson].

17. As to claim 18, Erickson as modified teaches storing access rights information of the user with the object, wherein the accessing the stored object comprises verifying the access rights of the user with the stored access rights information [col. 7, line 60 – col. 8, line 3 of Germscheid].

18. As to claims 19 – 21, they are rejected for the same reasons as claims 6, 8 and 9 above.

19. As to claim 22, this is an apparatus claim that corresponds to method claim 1; note the rejection to claim 1 above, which also meet this apparatus claim.

20. As to claim 27, this is rejected for the same reason as claim 7 above.

21. As to claim 28, Erickson as modified teaches a processor, a memory [host computer 1404; col. 26, lines 58 – 67 of Erickson], and a virtual machine executed by the processor from the memory, wherein the generating is performed by the virtual machine [col. 25, lines 58 – 67 of Erickson; examiner notes that a virtual machine is inherent to the Java environment].

22. As to claim 29, Erickson as modified teaches the accepting, the terminating, and the deleting are performed by the virtual machine [col. 25, lines 58 – 67 of Erickson], wherein the object is stored in the memory subsequent to the generating, and wherein, in the deleting, the object is deleted from the memory [col. 16, lines 1 – 6 of Germscheid].

Art Unit: 2194

23. As to claims 30 and 31, they are rejected for the same reasons as claims 8 and 9 above.

24. As to claims 32 – 34, these are system claims that correspond to method claims 10 – 12; note the rejections to claims 10 – 12 above, which also meet these system claims.

25. As to claim 40, this is rejected for the same reasons as claim 28 above.

26. As to claims 41 and 42, they are rejected for the same reasons as claims 8 and 9 above.

27. As to claim 43, this is a product claim that corresponds to method claim 1; note the rejection to claim 1 above, which also meet this product claim.

28. As to claim 45, this is rejected for the same reason as claim 7 above.

29. As to claim 46, this is rejected for the same reasons as claims 8 and 9 above.

30. As to claims 47 – 49, these are product claims that correspond to method claims 10 – 12; note the rejections to claims 10 – 12 above, which also meet these product claims.

31. As to claims 51 and 52, these are rejected for the same reasons as claims 16 – 18 above.

32. As to claim 53, Erickson as modified teaches the data representation language is eXtensible Markup Language (XML) [col. 25, lines 57-67 of Erickson] and the computer programming language is the Java programming language [col. 26, lines 1-16 of Erickson].

**33. Claims 3-5, 13-15, 24-26, 35-39 and 50 are rejected under 35 U.S.C. 103(a) as being unpatentable over Erickson and Germscheid in view of U.S. Patent No. 5,774,551 to Wu [cited in previous office action].**

34. As to claim 3, Erickson as modified does not teach accessing a client device by coupling an identification device to the client and terminating access by decoupling the identification device from the client device.

However, Wu teaches accessing a client device comprises the user coupling an identification device to the client device [authentication services 109 may include password or encrypted key based mechanisms such as...hardware/firmware based mechanisms, such as smart-card; col. 15, lines 54 – 65], wherein the identification device provides identification information of the user to the client device [the encrypted authentication tokens may be stored in a smart card, or other non-public storage facility; col. 10, lines 37 – 65], and wherein the termination the accessing comprises decoupling the identification device from the client device [system entry service 107 initiates a disconnect process, and handles the necessary physical disconnection and protocols for disconnecting from the system 100, Fig. 5; col. 19, line 57 – col. 20, line 9].

35. It would have been obvious to a person of ordinarily skilled in the art at the time of the invention to apply the teaching of accessing a client device by coupling an identification device to the client and terminating access by decoupling the identification device from the client device as taught by Wu to the invention of Erickson as modified because this allows any system entry service to be used transparently with any combination of account, password, session, or authentication services, including multiple instances of a given type of account management service and provides supports for unified login and logout with multiple authentication services [col. 6, lines 15 – 26 of Wu].

Art Unit: 2194

36. As to claim 4, Erickson as modified teaches the identification device is a smart card [the encrypted authentication tokens may be stored in a smart card, or other non-public storage facility; col. 10, lines 37 – 65 of Wu].

37. As to claim 5, Erickson as modified teaches the accessing a client device comprises the user logging on to the client device [Unified login is accomplished through a authentication token mapping process; col. 3, lines 55 – 67 of Wu] by providing user identification to the client device [the encrypted authentication tokens may be stored in a smart card, or other non-public storage facility; col. 10, lines 37 – 65 of Wu], and wherein the terminating the accessing comprises the user logging off the client device [user logouts 501 of the system entry service 107, either explicitly by invoking a specific method of the system entry service 107, or implicitly by shutting off the workstation or terminal the user is working on; col. 19, line 57 – col. 20, line 9 of Wu].

38. As to claims 13 – 15, they are rejected for the same reasons as claims 3 – 5 above. As to the motivation for combining Erickson and Germscheid with Wu, see the rejection to claim 3 above.

39. As to claims 24 and 25, these are apparatus claims that correspond to method claims 3 and 4; note the rejections to claims 3 and 4 above, which also meet these apparatus claims. As to the motivation for combining Erickson and Germscheid with Wu, see the rejection to claim 3 above.

40. As to claim 26, Erickson as modified teaches the device is further configured to accept user input [col. 4, lines 41 – 54 of Germscheid] to initiate the terminating the user access [unified logout process ensures that the user's authentication token and credentials are removed; col. 19, line 57 – col. 20, line 8 of Wu]. As to the motivation for combining Erickson and Germscheid with Wu, see the rejection to claim 3 above.

41. As to claims 35 and 36, these are system claims that correspond to method claims 13 and 14; note the rejections to claims 13 and 14 above, which also meet these system claims. As to the motivation for combining Erickson and Germscheid with Wu, see the rejection to claim 3 above.

42. As to claim 37, Erickson as modified teaches a memory [col. 26, lines 58 – 67 of Erickson], accept user input [col. 4, lines 41 – 54 of Germscheid] to terminate the access of the client device [col. 19, line 57 – col. 20, line 8 of Wu], and store the computer programming language object to the memory in response to the terminating access [col. 26, lines 21 – 30 of Erickson].

43. As to claims 38 and 39, they are rejected for the same reasons as claims 17 and 18 above.

44. As to claim 50, this is a product claim that correspond to method claim 13; note the rejection to claim 13 above, which also meet these product claims. As to the motivation for combining Erickson and Germscheid with Wu, see the rejection to claim 3 above.

### ***Conclusion***

45. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.



### CONTACT INFORMATION


46. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Li B. Zhen whose telephone number is (571) 272-3768. The examiner can normally be reached on Mon - Fri, 8:30am - 5pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, William Thomson can be reached on 571-272-3718. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Li B. Zhen  
Examiner  
Art Unit 2194

LBZ

  
WILLIAM THOMSON  
SUPERVISORY PATENT EXAMINER